

CREATION OF CUSTOMIZED TREES

PROVISIONAL APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/081,397, entitled "CREATION OF CUSTOMIZED TREES," filed on April 10, 1998, by Terry M. Bleizeffer et al., attorney's reference number ST9-98-029, which is incorporated by reference herein.

BACKGROUND OF THE INVENTION

1. Field of the Invention

10 This invention relates in general to computer-implemented systems, and, in particular, to creation of customized trees.

2. Description of Related Art

In conventional systems, a file system may be represented with a tree. A tree is a hierarchical structure that shows the relationship of one object to another. Each object is represented as a node in the tree. For example, a folder object may "contain" (i.e., 15 enclose) one or more folder objects, which contain one or more document objects. A user can access each object that is contained within another object by "drilling down" to that object. For example, to access a particular document object, the user first accesses the appropriate folder object that contains the document object. 20

However, in conventional systems, some of the objects in a tree are infrequently, or never, used by some users. In this case, the users may prefer not to see these particular objects. Instead, users may prefer to work with a small subset of the objects, rather than the entire tree. Additionally, in conventional systems, objects may not be ordered as a user desires. A user may wish to perform similar operations on several objects at the 25 same time. The objects may, however, be located in disparate places across different branches of the tree. There is no convenient mechanism for conveniently viewing an arbitrary set of objects and operating on some or all of them while they are being viewed.

There is a need in the art for improved trees.

SUMMARY OF A PREFERRED EMBODIMENT OF THE DISCLOSURE

To overcome the limitations in the prior art described above, and to overcome other limitations that will become apparent upon reading and understanding the present specification, an embodiment of the present invention discloses a method, apparatus, and
5 article of manufacture for computer-implemented creation of customized trees.

In accordance with the present invention, a customized tree is created in a computer from a original tree. The original tree contains objects from a data storage device connected to the computer. One or more objects are selected from the original tree
10 to be contained in the customized tree in response to user input. The selected objects are linked in a user-specified manner.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

15 FIG. 1 illustrates an exemplary computer hardware environment that could be used in accordance with an embodiment of the present invention;

FIG. 2 is an illustration of a tree based application within a window;

FIG. 3 is an illustration of a tree with branches and multiple objects;

FIG. 4 illustrates the tree illustrated in FIG. 3 with a pop-up that enables a user to
20 create a customized tree;

FIG. 5 illustrates a window displayed after a "Create customized tree . . ." action is selected;

FIG. 6 illustrates adding a database object to a customized tree;

FIG. 7 illustrates an Add Folder to Tree window that is displayed in response to
25 selection of the "Add folder to tree" action;

FIG. 8 illustrates the results of entering information into an Add Folder to Tree window of FIG. 7;

FIG. 9 illustrates another Add Folder to Tree window that is displayed in response to selection of the "Add folder to tree" action;

FIG. 10 illustrates the results of entering information into an Add Folder to Tree window of FIG. 9;

FIG. 11 illustrates a pop-up for adding another database folder object to a customized tree;

5 FIG. 12 illustrates another Add Folder to Tree window that is displayed in response to selection of the "Add folder to tree . . ." action;

FIG. 13 illustrates the results of entering information into an Add Folder to Tree window of FIG. 12;

FIG. 14 illustrates launching an open command from a parent application;

10 FIG. 15 illustrates the results of opening a customized tree;

FIG. 16 illustrates changing a customized tree;

FIG. 17 illustrates a Change Folder window that is displayed in response to a "Change . . ." action;

15 FIG. 18 is a flow diagram illustrating the steps performed by the customized tree creator to create a customized tree;

FIG. 19 is a flow diagram illustrating the steps performed by the customized tree creator to delete an object from a customized tree; and

FIG. 20 is a flow diagram illustrating the steps performed by the customized tree creator to modify an object in a customized tree.

20

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

In the following description of a preferred embodiment, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration a specific embodiment in which the invention may be practiced. It is to be understood that other embodiments may be utilized as structural changes may be made without departing from the scope of the present invention.

25

Hardware Environment

FIG. 1 is an exemplary hardware environment used to implement the preferred embodiment of the invention. The present invention is typically implemented using a computer 100, which generally includes, inter alia, a processor 102, random access memory

30

(RAM) 104, data storage devices 106 (e.g., hard, floppy, and/or CD-ROM disk drives, etc.), data communications devices 108 (e.g., modems, network interfaces, etc.), monitor 110 (e.g., CRT, LCD display, etc.), mouse pointing device 112 and keyboard 114. It is envisioned that attached to the computer 100 may be other devices such as read only
5 memory (ROM), a video card, bus interface, printers, etc. Those skilled in the art will recognize that any combination of the above components, or any number of different components, peripherals, and other devices, may be used with the computer 100.

The computer 100 operates under the control of an operating system (OS) 116, such as AIX™, OS/2™, WINDOWS NT™, WINDOWS™, UNIX™, etc. The operating system 116
10 is booted into the memory 102 of the computer 100 for execution when the computer 100 is powered-on or reset. In turn, the operating system 116 then controls the execution of one or more computer programs 118 by the computer 100. The present invention is generally implemented in these computer programs 118, which execute under the control of the operating system 116 and cause the computer 100 to perform the desired functions as
15 described herein.

The operating system 116 and computer programs 118 are comprised of instructions which, when read and executed by the computer 100, causes the computer 100 to perform the steps necessary to implement and/or use the present invention. Generally, the operating system 116 and/or computer programs 118 are tangibly embodied in and/or readable from a
20 device, carrier, or media, such as memory 102, data storage devices 106, and/or a remote device coupled to the computer 100 via the data communications devices 108. Under control of the operating system 116, the computer programs 118 may be loaded from the memory 102, data storage devices 106, and/or remote devices into the memory 102 of the computer 100 for use during actual operations.

25 Thus, the present invention may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" (or alternatively, "computer program product") as used herein is intended to

encompass a computer program accessible from any computer-readable device, carrier, or media. Of course, those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope of the present invention.

Those skilled in the art will recognize that the exemplary environment illustrated in FIG. 1 is not intended to limit the present invention. Indeed, those skilled in the art will recognize that other alternative hardware environments may be used without departing from the scope of the present invention.

The Creation of Customized Trees

10 In one embodiment, the present invention provides a customized tree creator 118 for creating a customized tree. The customized tree creator 118 works in conjunction with a tree-based application ("parent application") that represents objects with a tree. In one embodiment of the invention, the customized tree creator 118 adds functionality to the parent application by enabling a user to create a customized tree based on a tree
15 provided by the parent application. One skilled in the art would recognize that although the discussion herein illustrates the customized tree creator 118 working in conjunction with a parent application, the customized tree creator 118 is not necessarily, but may be, separate from the parent application.

The customized tree is typically generated from one or more other trees ("original trees"), which may be system generated trees or customized trees. The objects in the trees may represent a database management system and its objects, such as tables and indexes. However, one skilled in the art would recognize that the principles of the present invention can be applied to any type of object. The customized tree creator 118 enables users to create a customized tree that includes objects from multiple databases.
20 Additionally, the customized tree creator 118 enables users to create a customized tree to group together a subset of the objects in a customized tree that the users wish to work with. Moreover, the customized tree creator 118 enables users to modify the order of objects in the customized tree. Therefore, a customized tree can be created with a subset of objects, including objects from different databases, with the objects ordered as desired
25 by the user. The customized tree can be stored and used again, modified again, etc.
30

Additionally, the customized tree creator 118 enables users to organize and locate data. Furthermore, the customized tree creator 118 enables users to create a customized tree starting with any object in an original tree.

5 The customized tree creator 118 is particularly advantageous in that it enables a user to build a folder that contains objects, such as tables, from multiple databases. Once a folder has been created containing, for example, tables from two or three databases, when a user wishes to work with these tables, the user can access the tables using their corresponding nodes on the customized tree. On the other hand, in conventional systems, there may be multiple databases (e.g., database1 and database2) each with folders
10 containing tables. In conventional systems, users are unable to see all tables created across these two databases at the same time. Instead, in conventional systems, a user may be constrained to view each leaf node corresponding to a table separately.

The customized tree creator 118 is further advantageous in that, in a customized tree, the presentation of objects may be modified by the user. The customized tree creator
15 118 allows users to control presentation to suit their purpose, for example, for a particular view of a database. Users can order the objects of a database in a customized tree according to their data and what they want to manage. In one embodiment, there are no predefined limitations on the ordering. In another embodiment, there are some predefined limitations on the ordering, for example, preventing a table object from
20 containing a database object. On the other hand, in conventional systems, programmers, rather than users, often determine the presentation of objects.

Typically, a tree is a hierarchical structure that shows the relationship of one object to another. For example, a database object may be enclosed by another database object, and table objects may be enclosed by a database object. However, the tree
25 structure may not be strictly hierarchical, but, instead, may show one object enclosing another, enabling the drilling down to objects enclosed in other objects.

A customized tree can be used, in one embodiment, to provide a visual representation of a database. A user can select an object in the customized tree, for example, using a mouse to point at the object and clicking the mouse button. If a user
30 selects an object in the customized tree, a pop-up is displayed that lets a user select an action to perform on the selected object. Actions for a customized tree include, for

example, opening a customized tree, creating a customized tree, changing a customized tree, or removing part or all of a customized tree.

To remove a folder from a tree, a user can select one or more folder objects on the tree and choose the "Remove" action from the pop-up. A confirmation window appears asking the user if the user would like to remove the folder or folders selected. When the user confirms the remove action, the selected folder or folders are removed from the window displaying the tree. The changes then may be saved using, for example, a "Save" action (e.g., a Save action that would save the customized tree object to a file). An existing customized tree may be removed, For example, a user may use an operating system file system to locate a file containing the customized tree and delete the file. However, one skilled in the art would recognize that a user need not save the changes or could save the changes in a different manner and could remove the customized tree in a different manner.

Additionally, the customized tree creator 118 enables users to apply one or more actions against a folder object. That is, group actions may be applied to a folder object. For example, if a database node contained employee tables, then "dropping" the database object would "DROP" each of the employee tables contained in the database object. Similarly, other actions could be taken, such as changing privileges for a group of objects, replicating objects, defining user access, performing backups, or invoking other utility mechanisms.

Individual object actions can be applied to all of the contents of a folder object by choosing that action from the pop-up when a folder object is selected. For example, a user could choose a "Run Statistics" action after selecting a Tables folder object, and the "Run Statistics" action would be performed against all the tables in the Table folder. The user would not have to specify the individual tables. All of the specific actions for an individual table could also be available on the Table folder object. The same would be true for the other folder objects in the tree.

Customized trees would be beneficial to many users. For example, a senior database administrator who distributes work to junior database administrators may want to provide each junior database administrator with a separate customized tree, each with a particular group of objects that are relevant to the work of the junior database

administrator. In this way, customized trees can be used to compartmentalize the objects to which people have access. Additionally, customized trees provide convenience. For example, database administrators can provide customized trees to application programmers that include folders for databases needed by the application programmers, without including databases containing sensitive information (e.g., payroll tables).

In one embodiment of the invention, the Data Joiner product from IBM could be used to show databases from different platforms, for example, databases from Oracle or Sybase. With the ability to view databases from different platforms, a user is able to use the customized tree creator 118 to create a tree containing objects from the databases, from different platforms, that the user wishes to work with. Conventional systems do not provide the ability to group objects from databases across different platforms and view them together.

In one embodiment of the invention, the customized tree creator 118 enables any user to modify a customized tree. The databases typically have some security mechanisms (e.g., user identification checking), so that even if access to a customized tree is not restricted, a user may not be able to access particular databases because of that user's restricted access to those databases. In another embodiment of the invention, the customized tree creator 118 takes security measures to prevent modification of customized trees by one or more specified users. In particular, in a conventional system, a catalog which contains information about a database (e.g., tables, indexes, etc.) is used to set security restrictions. However, the customized tree creator 118 enables a user to define the security restrictions using the tree itself.

FIGS. 2-16 will be used to provide an example of customizing a tree. The customized tree creator 118 provides a series of graphical user interfaces to enable a user to easily customize a tree. The customized tree creator 118 creates, modifies, opens, etc. a tree in response to user input received from the graphical user interfaces.

The embodiment of the disclosure presented below illustrates an example of how the customized tree creator 118 may be utilized to produce a customized tree structure to suit the needs of a particular application. In FIGS. 2 and 3 illustrate "original" trees from which customized trees may be generated and illustrate some of the objects which may be present in databases. In FIGS. 4 and 5 of the example, a technique of creating a starting

point for a customized tree by selecting a "Create customized tree" action from a pop-up is illustrated. FIGS. 6 and 7 illustrate adding an element to the customized tree through the selection of an "Add folder to tree" action from a pop-up.

In particular, FIG. 7 illustrates a graphical user interface for adding filter criteria to an element, by which the user may restrict the elements depicted to those desired. In this manner, customized trees may be constructed of different elements, with different relationships between the elements, and views of the contents of the elements may be filtered to provide views consistent with the needs of users in various applications. The example further illustrates, in the remaining figures, the opening of customized trees that have been created and modification of those trees.

FIG. 2 is an illustration of a tree within a window 201, generated by a tree-based application. A database object may be represented using a tree, with folder objects in the tree representing the different databases and objects within these databases. The folder objects in the tree are shown in window 201. Within the window 201 is a subwindow 205 which displays a visual representation of the tree showing the objects contained in the tree. The systems folder 207 is shown as the root of the particular tree in this example.

FIG. 3 is an illustration of a tree with multiple branches and multiple objects, generated by a tree-based application. The tree-based application may include different databases, including a Universal Database 305 (illustrated as UDB1), a DataJoiner Server 307, DB2 for the OS/390 operating system 309 (illustrated as MVS1), or any number of database systems. Each database may contain other objects just as DSN1 313 contains the objects 315 as illustrated in the present example. As one skilled in the art will appreciate, there is no theoretical limit to how many objects or what types of objects may be contained within other objects.

FIG. 4 illustrates the tree illustrated in FIG. 3, within window 401, with a pop-up that enables a user to create a customized tree. Initially, a user selects a folder object from the original tree to be a starting object. In this illustration, object DSN1 403 is chosen as an initial starting point in creating a customized tree. The user can then choose an action to indicate a customized tree is to be created. In the present illustration, the customized tree creator 224 activates pop-up 405. The pop-up 405 contains the user

selection "Create customized tree" 407, which is shown highlighted. Additionally, the pop-up 405 includes an "Open new Control Center" 409 action, a "Change" 411 action, and a "Remove" 413 action. After activating the highlighted choice, "Create customized tree", in this example, the create customized tree process is activated.

5 FIG. 5 illustrates a "Customized Tree" window 501, generated by the customized tree creator 118, which is displayed after a "Create customized tree " action 407 is selected. In the window 501, the starting folder object is the DSN1 503, folder object selected in FIG. 4. The window is a duplicate of the window of FIG. 4, except that only the database DSN1 appears in the "Customized Tree" window 501. In this or a similar
10 manner, an initial starting point for a customized tree can be created.

After a starting point for the customized tree has been created, the process of transforming the starting point into a customized tree structure to fulfill the needs of a particular user can be undertaken. FIG. 6 illustrates an example augmentation of the initial starting point of a customized tree. In this case, a database object is added to the
15 customized tree. A user can add objects to the tree in the order the user wishes them to appear. For example, if a user wants to add a folder containing all the tables with the schema JONES that are in the DSN1 database, the user would select the DSN1 folder object 611. A pop-up 603 would then be activated by the customized tree creator 118. The pop-up 603 includes actions that can be taken for the selected folder object. For
20 illustration, the actions may include: "Add folder to tree" 605, "Change folder" 607, and "Remove folder" 609. In the illustration, the "Add folder to tree" 605 appears highlighted and is the action selected.

FIG. 7 illustrates an Add Folder to Tree window 701, generated by the customized tree creator 118, that is displayed in response to selection of the "Add folder to tree" 605
25 action. In the Add Folder to Tree window, an object type field 703 is provided. Object types may include tables, storage groups, tablespaces and a variety of other types. Those skilled in the art will recognize that a theoretically infinite variety of objects can be added. Additionally, in the present illustration of the invention, a folder label box 705 is provided. A folder label is a name for a folder, which will be displayed on the
30 customized tree. In FIG. 7, the folder label is "All JONES tables".

INS. A1 The folder label 705 can be used to distinguish between multiple folders enclosed in another folder. The folder label 705 may also provide an indication of the filter which is applied to the object. The Add Folder to Tree window 701 also enables a user to provide table filter information 707. The Illustrated fields 709, 711, and 713 illustrate example criteria for a filter. A criteria is entered in field 709, then field 711 is used to specify how the criteria is to be compared with the comparison values in field 713. In the next line, the Schema 715 is checked via the technique of comparison chosen 717, for equality, to the comparison value Jones in field 719 which is set equal to JONES. The comparison criteria may be ">", "<", "<=" or any other of those well known in the art. In addition, compound filters may be created through Boolean logic comparisons, such as 715.

Filter criteria is used to determine objects to be contained within one of the objects of the customized tree, for example, the root object. For example, filter criteria may be used to determine which tables are to be contained in the DSN1 folder object. Once a filter has been created, if the tables upon which the filter is applied change, the filter automatically modifies the customized tree to reflect the changed tables upon receiving, for example, a REFRESH command, which refreshes the screen. In one example of a selection criteria technique, if a folder is created with Jones tables according to this example, when additional Jones tables are added, the additional Jones tables are included in the folder. However, one skilled in the art would recognize that other selection criteria techniques may be used with the principles of the present invention. When a user is satisfied with the filter which has been created, the user may lock in the values by clicking "OK" 723.

In FIG. 8 the Customized Tree window 801, generated by the customized tree creator 118, illustrates the results of entering information into an "Add Folder to Tree" window of FIG. 7. In FIG. 8, the original folder DSN1 803, is still present in the "Customized Tree" window 801 and a new folder object 805 is contained by the DSN1 folder object. The tree has changed to include the folder 805 with the label "All JONES tables" 807. The new folder appears in subwindow 817 below DSN1 803. The "All JONES tables" label 807 is highlighted as indicative that it is the present item in subwindow 817, which is selected.

INS. A2 The folders of the tree in the subwindow 817 on the left are considered containers. When the All JONES tables folder is selected (e.g., by clicking on it), the subwindow 819 displays the tables 815 contained in that folder. For example, the All JONES Tables folder contains multiple tables, including, for example, a STAFF table. Fields 811, and 5 813 are labels for the content of the folder 805. Horizontal scroll bar 819 is included so that all of the contents of the elements contained in folder 805 may be displayed even if the elements occupy a larger area than can be displayed in window 821.

FIG. 9 illustrates another "Add Folder to Tree" window 901, generated by the customized tree creator 118, that is displayed in response to a selection of the "Add folder to tree" action. In this example a user has decided to add a folder object to the 10 customized tree. As shown in field 903 the object type to be added is a database which will be labeled EMPLOYEE, as shown in field 905, to the customized tree. The user highlights the DSN1 folder 803 and activates the pop-up. The user selects the "Add folder to tree" action from the pop-up. When the "Add folder to tree" action is activated 15 the "Add Folder to Tree" window 901 is displayed by the customized tree creator 118. The user indicates that the object type is a database, as shown in field 903. In this Add Folder to Tree window, an object of type Databases is to be added, with label "EMPLOYEE" as entered in field 905. The filter criteria of Name, which is entered in field 907, is to be compared for equality, as specified in field 907, to EMPLOYEE as 20 specified in field 913. When a user clicks "OK" 915 in the Add Folder to Tree window 901, the customized tree creator adds the EMPLOYEE database the customized tree.

FIG. 10 illustrates the results of entering information into an "Add Folder to Tree" window 901 of FIG. 9. In FIG. 10, a new folder object, "Databases" 1003 is displayed, which contains an EMPLOYEE database 1005. The label EMPLOYEE is shown as 25 selected by the highlight 1007.

FIG. 11 illustrates a pop-up in the customized tree window 1101 for adding another database object to the present example of a customized tree. To add a table folder under the EMPLOYEE database folder, a user activates the pop-up 1103 and selects the "Add folder to tree" 1105 action from the pop-up, generated by the 30 customized tree creator 118.

INS. A3 > FIG. 12 illustrates another "Add Folder to Tree" window 1201, generated by the customized tree creator 118, that is displayed in response to selection of the "Add folder to tree" action 1105. In this Add Folder to Tree window, an object of type "Tables" is entered in field 1203. The folder label of "JONES tables in EMPLOYEE" is created by entering "JONES tables in EMPLOYEE" in field 1207. The filter is Schema 1209 is chosen to be "Equal to" 1211 JONES, as entered in field 113. The user then clicks the "OK" button 1215 to accept the information as entered in window 1201.

FIG. 13 illustrates the results of entering information into an Add Folder to Tree window 1201 of FIG. 12. When the user clicked the "OK" button 1215 of the Add Folder to Tree window 1201, a Tables folder 1307 labeled "JONES tables in EMPLOYEE" 1309 is added to the tree under the EMPLOYEE database 1305. The customized tree creator 118 adds Tables folder 1307 under the EMPLOYEE database 1305, and not under DSN1 1303, because the user invoked the "Add folder to tree" action from the pop-up while the EMPLOYEE database folder 1305 was highlighted (i.e. selected). The list on the subwindow 1311 shows those individual tables 1313 in the EMPLOYEE database 1305 that have the schema of JONES 1315. This customized tree could then be saved.

INS. A4 > FIG. 14 illustrates launching an open command from a parent application. To invoke an existing customized tree, a user selects "Open" 1405 from the window 1403 pull down menu. A list of existing trees follow in a cascade 1407, and the user can select one from the cascade list. In the present example Bob's tree, 1409 has been selected as indicated by the highlight. In this example, the customized tree may also be invoked from a customized tree window, for example customized tree windows as shown in FIGS. 5, 6, 8, 10, 11, 13, by choosing the "Open" action and then selecting one of the available objects to open. The customized tree is illustrated as being opened from the "Control Center" window only as a matter of convenience, and a customized tree may be opened from a "Customized Tree" or "Control Center" window with equivalent results.

FIG. 15 illustrates the results of opening a customized tree. In FIG. 15, "Bob's Tree" is displayed in the title line of the opened window 1501 in response to the "Open" and selection of Bob's Tree in FIG. 13. As the result of opening Bob's Tree, Bob's Tree is also displayed 1505 in the left subwindow 1507.

Additionally, to save a new customized tree or changes to an existing customized tree, a user would select "Save" 1411 or "Save as" 1413 from the pull down menu, as illustrated in FIG. 14. The tree would then be saved as a file either on a user's workstation or on a server.

5 FIG. 16 illustrates a "Change Folder" action 1603 in pop-up 1605, generated by the customized tree creator 118. To change a customized tree, a user selects a folder object (e.g. 1607) on the tree. The subwindow 1609 shows the contents of the selected object 1607. The user activates the pop-up 1605, generated by the customized tree creator 118, and selects the "Change folder" action 1603 from the pop-up 1605 for the
10 selected folder object 1607.

INS. 15 FIG. 17 illustrates a Change Folder window 1701, generated by the customized tree creator 118, that is displayed in response to a "Change" action. The Change Folder window 1701 is the same as the Add Folder to Tree window 1201, except that the window reflects the selections previously made. In the present example the object type
15 field is table 1703, and the Folder label for object field 1705 is "ALL JONES Tables". The filtering criteria 1707, 1709, 1711, 1713, 1715, and 1719 may be changed by the user if a different criteria is desired. The user makes changes desired and those changes take effect when the user clicks on the "OK" button 1721. The changes would have to be saved, for example, with the "Save" action 1311 and the contents of the object which had
20 been selected 1607, may change as appropriate. The changed contents will then be reflected in the contents of the folder as displayed in the right subwindow when the object is selected, e.g. subwindow 1609.

FIG. 18 is a flow diagram illustrating the steps performed by the customized tree creator 118 to create a customized tree. In Block 1800, the customized tree creator 118
25 selects an object on an original tree to be contained in the customized tree in response to user input. In Block 1802, customized tree creator 118 receives a request to add a folder to the customized tree for the selected object. In Block 1804, the customized tree creator 118 displays an "Add Folder to Tree" window in response to the request. In Block 1806, the customized tree creator 118 creates a filter for the selected object in response to user
30 input from the displayed window. The filter specifies a selection criteria to select objects to be contained in the selected object in the customized tree. In Block 1808, the

customized tree creator 118 applies the filter to create the customized tree with the selected object and the objects to be contained within the selected object.

FIG. 19 is a flow diagram illustrating the steps performed by the customized tree creator 118 to delete an object from a customized tree. In Block 1902, the customized tree creator 118 receives a user's selection of an object to be deleted. In Block 1904, the customized tree creator 118 receives the user's selection of the "Remove folder" choice 1107 from the pop-up, as illustrated in FIG. 11. In Block 1906, the customized tree creator 118 removes the selected object from the tree.

FIG. 20 is a flow diagram illustrating the steps performed by the customized tree creator 118 to modify an object in a customized tree. In Block 2002, the customized tree creator 118 receives a user's selection of an object to be deleted. In Block 2004, the customized tree creator 118 receives the user's selection if the "Change folder" choice 1603 from the pop-up, as illustrated in FIG. 16. In Block 2006, the customized tree creator 118 modifies the selected object.

Conclusion

This concludes the description of the preferred embodiment of the invention. The following describes some alternative embodiments for accomplishing the present invention. For example, any type of computer, such as a mainframe, minicomputer, or personal computer, or computer configuration, such as a timesharing mainframe, local area network, or standalone personal computer, could be used with the present invention.

Those skilled in the art will recognize that the above description of an embodiment of the invention is merely an example of a single embodiment of the invention meant to serve as an illustration of one example of how the invention might be implemented. One skilled in the art will recognize that the concepts and ideas of the invention may be embodied in a virtually limitless combination of user screens, input mechanisms, drop down menus, pointing mechanisms, control fields, command lines and the like, which are input and control mechanisms for computers and are well known in the art. Those skilled in the art will not construe the above described menu system as constraining the ideas and claims of the invention, but merely depicting an implementation which may vary according to application and individual preferences.

The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims
5 appended hereto.

067207 342,84260